







Comparative Evaluation of Hyperparameter Tuning Methods in Crop Yield Prediction for Enhancing Food Security



S Jayanthi^{1*}, T. Suvarna Kumari², Srujana Inturi², B. Nathan³, M. A. Josephine Sathya⁴,
Karthik Karmakonda⁵

¹ Department of Artificial Intelligence and Data Science, Faculty of Science and Technology (ICFAITech), The ICFAI Foundation for Higher Education, Hyderabad 501203, India

² Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, India

³ Department of Computer Science and Engineering, Dhaanish Ahmed Institute of Technology, Coimbatore 641105, India

⁴ Department of Computer Science and Applications, Christ Academy Institute for Advanced Studies, Bangalore 560083, India

⁵ Department of Computer Science and Engineering, CVR College of Engineering, Hyderabad 501510, India

Corresponding Author Email: drsjayanthicse@gmail.com

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijse.160301>

ABSTRACT

Received: 31 December 2025

Revised: 12 February 2026

Accepted: 17 March 2026

Available online: 31 March 2026

Keywords:

hyperparameter tuning, crop yield prediction, GridSearchCV, RandomizedSearchCV, Optuna, nested cross-validation, Random Forest, XGBoost

An accurate crop yield prediction (CYP) system is crucial for enhancing agricultural practices, ensuring food security, and reducing harvest risks. Machine learning (ML) models are effective for predictive tasks, but their performance depends heavily on appropriate hyperparameter tuning. Studies comparing the predictive accuracy along with computational efficiency of hyperparameter tuning methods remain limited. We benchmark three hyperparameter tuning methods, GridSearchCV (GSCV), RandomizedSearchCV (RSCV), and Optuna (OPT), across five ML models, namely Decision Tree (DT), Random Forest (RF), Gradient Boosting (GB), XGBoost (XGB), and LightGBM (LGBM). We employ a nested cross-validation framework for unbiased results, along with statistical significance testing and SHapley Additive exPlanations (SHAP)-based interpretability analysis. The experiments are conducted on a CYP dataset comprising agronomic and climatic variables of 101 countries. OPT shows performance comparable to GSCV across multiple models. It is evidenced by the RF model which attained the lowest Mean Squared Error (MSE) of 1.08×10^7 , Root Mean Squared Error (RMSE) of 3280.77, Mean Absolute Error (MAE) of 432.30, and R^2 of 0.9987, under both OPT and GSCV. However, OPT is associated with higher computational cost, whereas RSCV improves efficiency by reducing tuning time, with a modest trade-off in accuracy. Overall, no single tuning method consistently outperforms the others across all models. A tuning strategy should be chosen after careful consideration of trade-offs among stability, processing time, and resource availability restrictions. This study demonstrates that hyperparameter optimization strategies are pivotal in determining the performance, computational efficiency, and practical applicability of ML models, thereby enabling more reliable agricultural decision-making and policy support.

1. INTRODUCTION

Changing weather patterns and the impacts of climate change on soil health across the globe have made accurate crop yield forecasting crucial for ensuring global food security and reducing harvest risks [1-4]. It helps farmers and policymakers to make informed decisions on land use, irrigation, and supply chain management [5-8]. Machine learning (ML) models have contributed to crop yield prediction (CYP) by analyzing diverse patterns in agricultural data [9-11]. Selecting hyperparameters of ML models is critical in optimizing their performance. Thus, their reliability depends on suitable hyperparameter tuning techniques. Poorly tuned models can lead to the design of suboptimal prediction models, ineffective resource utilization, and high computational costs [11-13].

GridSearchCV (GSCV) [14], RandomizedSearchCV

(RSCV) [15], and Optuna (OPT) [16] are widely used techniques for hyperparameter tuning in ML models [17-19], yet they differ significantly in computational efficiency, search strategy, and predictive performance. GSCV examines every combination of parameters to ensure optimal tuning, but its exhaustive nature results in high computational costs [14]. RSCV improves computational efficiency by randomly sampling a fixed number of hyperparameter combinations from the search space. This avoids the exhaustive evaluation of all possible configurations [15].

OPT uses Bayesian optimization and pruning techniques to effectively explore the search space. In each trial, it selects effective hyperparameter configurations instead of relying on random or exhaustive search through Bayesian optimization. Pruning strategies help to terminate underperforming trials at early stages and thereby reduce computation overhead [17].

Despite the growing adoption of these techniques, existing studies mainly focus on predictive accuracy. The trade-offs between accuracy, computation cost, and model stability introduced by hyperparameter tuning are overlooked. Moreover, there is a lack of controlled benchmarking studies on evaluating these tuning strategies. Additionally, there is a limited comparative study of these methods in the agriculture domain. In particular, with limited insight into the interplay between their predictive performance and computational expense.

We empirically investigate the efficacy of GSCV, RSCV, and OPT for hyperparameter optimization in five ML models, namely Decision Tree [20], Random Forest [21], Gradient Boosting (GB) [22], XGBoost (XGB) [23], and LightGBM (LGBM) [24] with a CYP dataset covering significant agronomic variables such as harvested area, crop type, pesticide use, rainfall, and temperature. We performed this study intensively with nested cross-validation to avoid bias in model selection.

The major objectives of this study are:

- Carried out a comparative study of GSCV, RSCV, and OPT in optimizing ML models' parameters
- Expounded optimal parameters for the studied ML models
- Empirically evaluated predictive performance and tuning time using nested cross-validation
- Explored the impact of meteorological, agronomic, and environmental features on crop yield

The findings of this study will guide users to select the most appropriate hyperparameter optimization methods based on task requirements, as well as time and resource constraints in agricultural applications. The rest of the paper is organized in the following manner: Section 2 examines existing literature on ML-based CYP and hyperparameter tuning schemes. Section 3 discusses the utilized dataset and its preprocessing activities, modeling methods, and processes. Section 4 details the hyperparameter search space and evaluation metrics. Section 5 deliberates the experimental results and performance analysis. The last section deliberates on the important findings, drawbacks, and recommendations for future studies.

2. RELATED WORK

CYP is a fundamental challenge in agricultural risk management and food security. ML models have been widely adopted due to its ability to model non-linear interactions among climatic, agronomic, and environmental variables [25]. Since farming features are often affected by environmental factors, soil conditions, and plant types, exploring hyperparameter tuning methods and their comparative evaluation is mandatory. Recent literature consistently identifies ensemble-based architectures, such as RF and XGBoost, as the benchmarks for predictive accuracy due to their resilience against the noise and heteroscedasticity common in multi-regional agricultural data [26, 27].

As noted earlier, the efficacy of ensemble models is intrinsically tied to their hyperparameter configurations, which govern the equilibrium between model complexity and generalization capacity [9]. In agricultural applications, hyperparameter tuning has received comparatively limited attention. Several studies have applied ML models for CYP using climatic and soil features with the aim of achieving high predictive accuracy [28-29]. More recent works have

incorporated optimization techniques, including Bayesian optimization frameworks such as OPT, RSCV, GSCV to improve model performance and interpretability [30, 31].

Despite these advances, existing studies typically employ a single optimization strategy or focus on specific model configurations, without systematically comparing alternative tuning methods under consistent experimental conditions. Furthermore, most works evaluate performance primarily in terms of predictive accuracy, with limited consideration of computational cost, tuning time, and efficiency factors that are critical for real-world deployment in agricultural decision-support systems [13, 16, 17].

A few studies have shown that GSCV can improve yield forecasting accuracy by optimizing Support Vector Machines [12]. However, GSCV is computationally intensive for exhaustive searching and thus less scalable in large datasets. Contrarily, RSCV is more scalable as it randomly samples a combination of hyperparameters, but may give more variable results since RSCV is faster but less consistent in results.

2.1 Research gap and motivation

While hyperparameter optimization has seen significant technical growth, three critical limitations persist in the current literature of CYP:

- Most studies apply a single optimization method without a comparative evaluation against alternative methods.
- Second, performance evaluation is predominantly accuracy-driven and often overlook the critical trade-offs between predictive precision, computational overhead, and tuning latency.
- There is limited evidence on whether advanced tuning methods like Optuna maintain stable feature attributions compared to traditional methods when processed through XAI frameworks.

To address these limitations, this study introduces a unified benchmarking framework that systematically evaluates GSCV, RSCV, and OPT across multiple ML models under consistent experimental conditions. By integrating nested cross-validation and SHAP (SHapley Additive exPlanations) based interpretability [31, 32], this work provides a comprehensive analysis of the trade-offs among predictive accuracy, computational efficiency, and model stability. It has direct relevance for food-security-oriented decision support.

3. METHODOLOGY

This research adopts a systematic workflow that comprises data preprocessing, model selection, hyperparameter tuning, and comprehensive performance evaluation, as demonstrated in Figure 1.

3.1 Dataset description

The study uses publicly available data from the Food and Agriculture Organization (FAO) and the World Bank. These datasets are intended to support agricultural risk management and forecasting while providing comprehensive information for CYP and early warning systems for food security. The dataset was carefully prepared and included significant details on agronomic, climatic, and chemical factors affecting agricultural output. These factors are very crucial in shaping

the ML models in CYP. The data set includes 28242 instances gathered from 1990 up to 2023 over 101 geographical regions, including different agro-climatic zones. Table 1 presents the dataset attributes and their relevance to CYP.

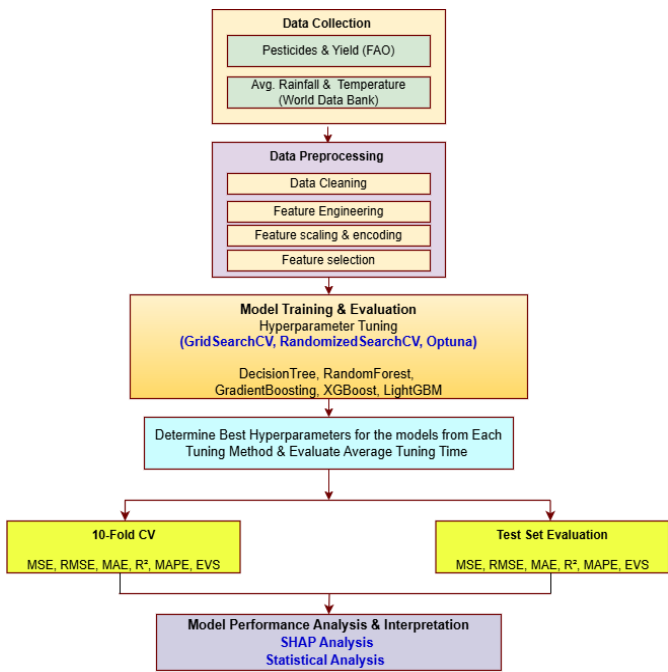


Figure 1. Workflow of the proposed ML-based crop yield prediction (CYP) framework

Table 1. Dataset attributes and their descriptions

Attributes	Description
Input Features	
Area	The geographical area in which the crop is grown.
Item	Type of crop cultivated.
Year	Year in which the data were gathered.
Pesticide Use (tonnes)	Amount of pesticides used.
Average Rainfall (mm/year)	Precipitation levels influencing soil moisture and irrigation needs.
Average Temperature (°C)	Climatic factors influencing plant growth.
Target Variable	
Crop Yield (hg/ha yield)	Productivity of crops in hectograms per hectare.

Figure 2 illustrates the variation in normalized rainfall, pesticide usage, temperature, and crop yield across different geographical areas. The plot shows that crop yield varies considerably across regions, even when environmental inputs exhibit similar magnitudes. It can be interpreted that higher rainfall and temperature do not consistently coincide with higher crop yield in several regions. Conversely, some regions achieve relatively higher yields under moderate levels of rainfall, temperature, and pesticide usage. It also shows that low rainfall can still exhibit moderate crop yields. This may reflect the influence of factors like irrigation, good soil conditions, and effective crop management.

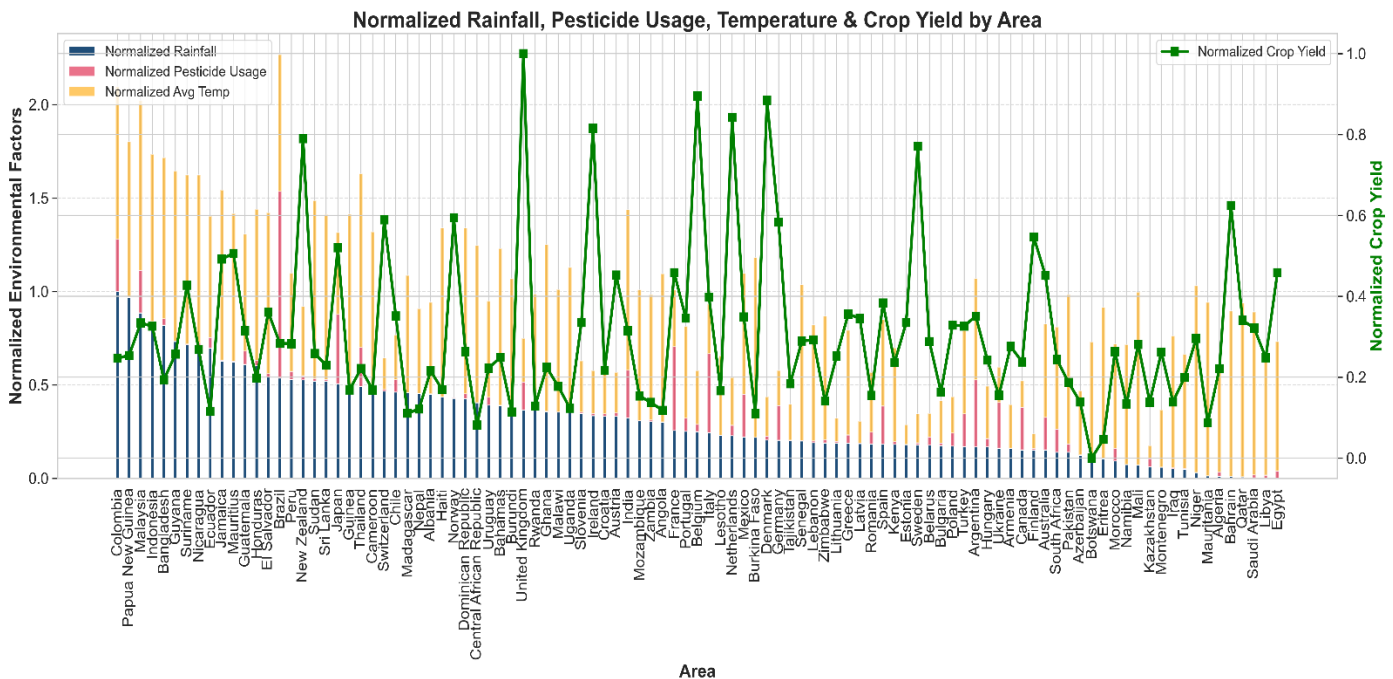


Figure 2. Relationship between rainfall, pesticide usage, temperature, and crop yield by area

3.2 Data preprocessing and feature engineering

We encoded the categorical variables (Area and Item) into numerical representations suitable for processing using label encoding. This process enables tree-based models not to be sensitive to ordinal relationships in the encoded features. To address skewness in environmental variables, pesticide usage was transformed using a \log_{1p} function, which safely handles

zero values. Missing values, if any, were treated using median imputation to ensure data consistency.

We applied RobustScaler to normalize input features by minimizing the effect of outliers using median and interquartile range. This feature scaling facilitates model stability and convergence during training. Before performing the prediction, we explored several domain-informed features. These features include indicators that show changes over time,

such as last year's yield, the yield from two years ago, and the average yield over the last three years. These indicators showcase the inherent trends more clearly. We examined the interaction between rainfall and temperature, as well as the impact of pesticides on temperature, to understand the relationships among various agroclimatic factors. We calculated a detrended temporal feature to measure how each crop and region changes over time. These features, which capture both time-related patterns and complex interactions, were used to improve predictions of crop yields.

The dataset was sorted chronologically and split into training (80%) and test (20%) sets, with earlier years allocated to training and later years to testing. Model evaluation and hyperparameter optimization are performed using a nested cross-validation scheme. The outer loop includes 5-fold cross-validation to ensure an unbiased evaluation of the models. The inner loop employs 5-fold cross-validation with random shuffling for the purpose of effectively tuning the hyperparameters of the three methods. A fixed random seed (42) is used consistently for model initialization across all three tuning methods to ensure reproducibility. GSCV performs an exhaustive search over predefined grids, whereas RSCV and Optuna explore the search space using 50 iterations. Optuna uses the Tree-structured Parzen Estimator (TPE), a sequential Bayesian optimization algorithm that models the objective function to guide the selection of promising hyperparameter configurations. We did not apply any no pruning strategy; all trials are run to completion. Each configuration is trained using a fresh model instance to ensure independence between evaluations. All preprocessing steps are applied within the training folds and consistently extended to validation and test data to avoid information leakage.

3.3 Machine learning models

ML models were chosen for investigation based on their theoretical rationale and ability to handle non-linear interactions in the agricultural dataset to identify the most reliable and accurate CYP.

The DT model was chosen because of its transparency and simplicity. It recursively divides the feature space based by minimizing the variance of the target variable within each node, typically using Mean Squared Error (MSE). To mitigate overfitting without sacrificing interpretability, we included the RF model. This method combines the predictions from several DTs, each trained on a bootstrapped sample of the data. The final yield prediction is obtained by calculating the arithmetic mean of the individual tree outputs, as shown in Eq. (1):

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(X) \quad (1)$$

where, $h_t(X)$ represents prediction from a single tree t , and T represents the number of trees. Since RFs effectively reduce variance and capture complex feature interactions, they are robust in handling large datasets.

GB was incorporated to increase the predictive power, which sequentially corrects errors in previous iterations in regression tasks. Weak learners are trained iteratively to reduce a loss function with Eq. (2):

$$F_m(X) = F_{m-1}(X) + \gamma h_m(X) \quad (2)$$

where, $F_m(X)$ is the boosted model at iteration m , h_m is the weak learner, and γ is the learning rate controlling the contribution of each learner. GB works very well in most cases when the relations are highly non-linear.

XGB was applied because of its computational efficiency. In this regression context, it applies regularization to reduce overfitting, making it an optimized version of gradient boosting. Its optimization function includes both loss minimization and model complexity and is computed using Eq. (3):

$$L(\theta) = \sum_{i=1}^n l(t_i, \hat{t}_i) + \lambda \sum_j \|w_j\|^2 \quad (3)$$

In the above equation, t_i and \hat{t}_i denote the actual and predicted crop yield for observation i , respectively. $l(t_i, \hat{t}_i)$ and $\lambda \sum_j \|w_j\|^2$ correspond to the loss function in MSE and the regularization term, respectively. XGB exploits parallel computation and tree-pruning techniques which makes XGB an efficient and scalable algorithm.

We employed LGBM, a high-performance gradient boosting tool to further enhance the performance. Unlike the GB, it performs histogram-based feature binning to speed up training with the same level of accuracy.

The calculation of split gain in LGBM is performed using Eq. (4):

$$G = \frac{(\sum_{i \in L} gain_i)^2}{|L|} + \frac{(\sum_{i \in R} gain_i)^2}{|R|} - \frac{(\sum_{i \in P} gain_i)^2}{|P|} \quad (4)$$

where, $gain_i$ indicates the gradient of the loss function while L , and R are left and right child nodes, respectively, and P is the parent node. This model is suitable for large scale data because it efficiently deals with categorical features and uses less memory.

4. HYPERPARAMETER TUNING STRATEGIES

This section presents the details of the tuning methods, the parameter search space for each model, and the best parameter identified by the tuning methods.

GSCV performs optimization by evaluating every possible configuration of hyperparameters for each given model, where h_i denotes the i -th hyperparameter with m_i possible values, the total number of evaluations is given by Eq. (5):

$$N_{Total} = \prod_{i=1}^n m_i \quad (5)$$

GSCV exhaustively explores the predefined search space and identifies the best-performing hyperparameter configuration within that space, at the cost of significant computational resources. This study focused on essential hyperparameters such as the number of estimators, learning rate, tree depth, and regularization parameters.

RSCV addresses inefficiencies incurred in GSCV by randomly sampling k configurations from the hyperparameter space. This technique is effective when certain hyperparameters have less impact, more search space, and efficient sampling is desired.

The hyperparameter optimization performed by OPT is based on Bayesian optimization, with the TPE. In every iteration, OPT samples a new hyperparameter configuration θ

according to expected improvement (EI), using Eq. (6):

$$EI(\theta) = E[\max(f(\theta) - f^*, 0)] \quad (6)$$

where, $f(\theta)$ is the model's performance with hyperparameters, θ and f^* is the best observed performance. Also, early stopping support provided by OPT helps to save computational time as it eliminates failing trials before completion. This form of optimization achieves efficient convergence with minimum computational costs.

4.1 Hyperparameter search space

The hyperparameter ranges searched for every model are shown in Table 2.

The optimal hyperparameters identified for each selected ML models are summarized in Table 3.

Table 2. Hyperparameter search space

Model	Parameter	Range
Decision Tree	max_depth	{None, 5, 10, 20}
	min_samples_split	{2, 5, 10}
	n_estimators	{50, 100, 150, 200}
Random Forest	max_depth	{None, 5, 10, 20}
	min_samples_split	{2, 5, 10}
	min_samples_leaf	{1, 2, 4}
	n_estimators	{50, 100, 150, 200}
Gradient Boosting	learning_rate	{0.05, 0.1, 0.2}
	max_depth	{3, 5, 7}
	n_estimators	{50, 100, 150, 200}
XGBoost	learning_rate	{0.05, 0.1, 0.2}
	max_depth	{3, 5, 7}
	n_estimators	{50, 100, 150, 200}
LightGBM	learning_rate	{0.05, 0.1, 0.2}
	num_leaves	{31, 50, 100}

Table 3. Summary of optimal hyperparameter configurations by model and tuning technique

Model	Tuning	Number of Estimators	Learning Rate	Max. Depth	Min. Samples per Split	Min. Samples per Leaf	Number of Leaves	Avg. Tuning Time (s)	Avg. Training Time (s)
Decision Tree	GSCV	–	–	None	5	–	–	1.20	0.166
	RSCV	–	–	None	5	–	–	1.09	0.172
	OPT	–	–	None	5	–	–	6.04	0.170
Random Forest	GSCV	150	–	None	2	1	–	180.03	16.47
	RSCV	150	–	None	2	2	–	64.06	17.33
	OPT	150	–	None	2	1	–	228.16	16.66
Gradient Boosting	GSCV	150	0.2	7	–	–	–	130.20	16.21
	RSCV	150	0.1	7	–	–	–	59.38	15.89
	OPT	150	0.1	7	–	–	–	202.77	14.03
XGBoost	GSCV	150	0.2	7	–	–	–	8.18	0.281
	RSCV	150	0.1	7	–	–	–	3.37	0.286
	OPT	150	0.2	7	–	–	–	12.98	0.220
LightGBM	GSCV	150	0.2	–	–	–	100	55.62	0.666
	RSCV	150	0.1	–	–	–	100	21.18	0.466
	OPT	100	0.2	–	–	–	100	78.32	0.419

4.2 Evaluation metrics for rigorous predictive model assessment

We calculated MSE to compute the average squared difference between predicted and actual crop yields using Eq. (7):

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2 \quad (7)$$

where, t_i and \hat{t}_i are the actual and predicted crop yields, respectively, and n is the number of data points. Since MSE is quadratic in nature, it penalizes more significant errors more heavily, making it sensitive to outliers. We also computed Root Mean Squared Error (RMSE) to estimate errors in the original units of the target variable using Eq. (8).

$$RMSE = \sqrt{MSE} \quad (8)$$

We also computed Mean Absolute Error (MAE) as it measures the average absolute difference between actual and predicted outputs using Eq. (9).

$$MAE = \frac{1}{n} \sum_{i=1}^n |t_i - \hat{t}_i| \quad (9)$$

This metric remains a robust measure of prediction error which is less sensitive to outliers. Then, we computed Mean Absolute Percentage Error (MAPE) to assess relative prediction accuracy using Eq. (10).

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{t_i - \hat{t}_i}{t_i} \right| \quad (10)$$

This metric helps compare model performance on different datasets with varying scales of yields but is not defined if any t_i is 0. To evaluate the goodness of fit, we computed R^2 score, which measures the proportion of variation in yield based on input features using Eq. (11).

$$R^2 = 1 - \frac{\sum_{i=1}^n (t_i - \hat{t}_i)^2}{\sum_{i=1}^n (t_i - \bar{t})^2} \quad (11)$$

R^2 score can take values less than 0, with higher values (closer to 1) indicating better model performance. We also applied Explained Variance Score (EVS) to measure how well a model explains the variance of the target variable, from $-\infty$ to 1, with 1 representing perfect prediction using Eq. (12).

$$EVS = 1 - \frac{Var(t - \hat{t})}{Var(t)} \quad (12)$$

5. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

5.1 Experimental setup

The study was performed on a 12th generation Intel Core i9-12900 with a 64-bit Windows 11 Pro operating system. The software setup was created with a Python 3.11 environment and the following libraries: NumPy, Pandas, Seaborn, Matplotlib, SciPy, StatsModels, Scikit-Posthocs, Optuna, and Scikit-learn. We used GSCV and RSCV to tune the hyperparameters. The models used to investigate the tuning methods are DT, RF, GB, XGB, and LGBM.

5.2 Accuracy-based comparison

To ensure the reliability of the models, the predictive performance of the evaluated models during the cross-validation phase is documented in (Figure 3 and Table 4) and subsequently verified against the independent test set (Figure 4 and Table 5). RF has the lowest error rates in both cross-validation and test sets. It achieved a test RMSE of 3280.77 and R^2 of 0.9987 under both GSCV and OPT, indicating strong generalization. GB and XGB work well with GSCV and OPT. However, their performance drops when using RSCV. For example, the RMSE of XGB increases from 4985.67 (GSCV) to 6178.80 (RSCV). It represents an approximate 23.9% increase and evinces its sensitivity to suboptimal hyperparameter configurations.

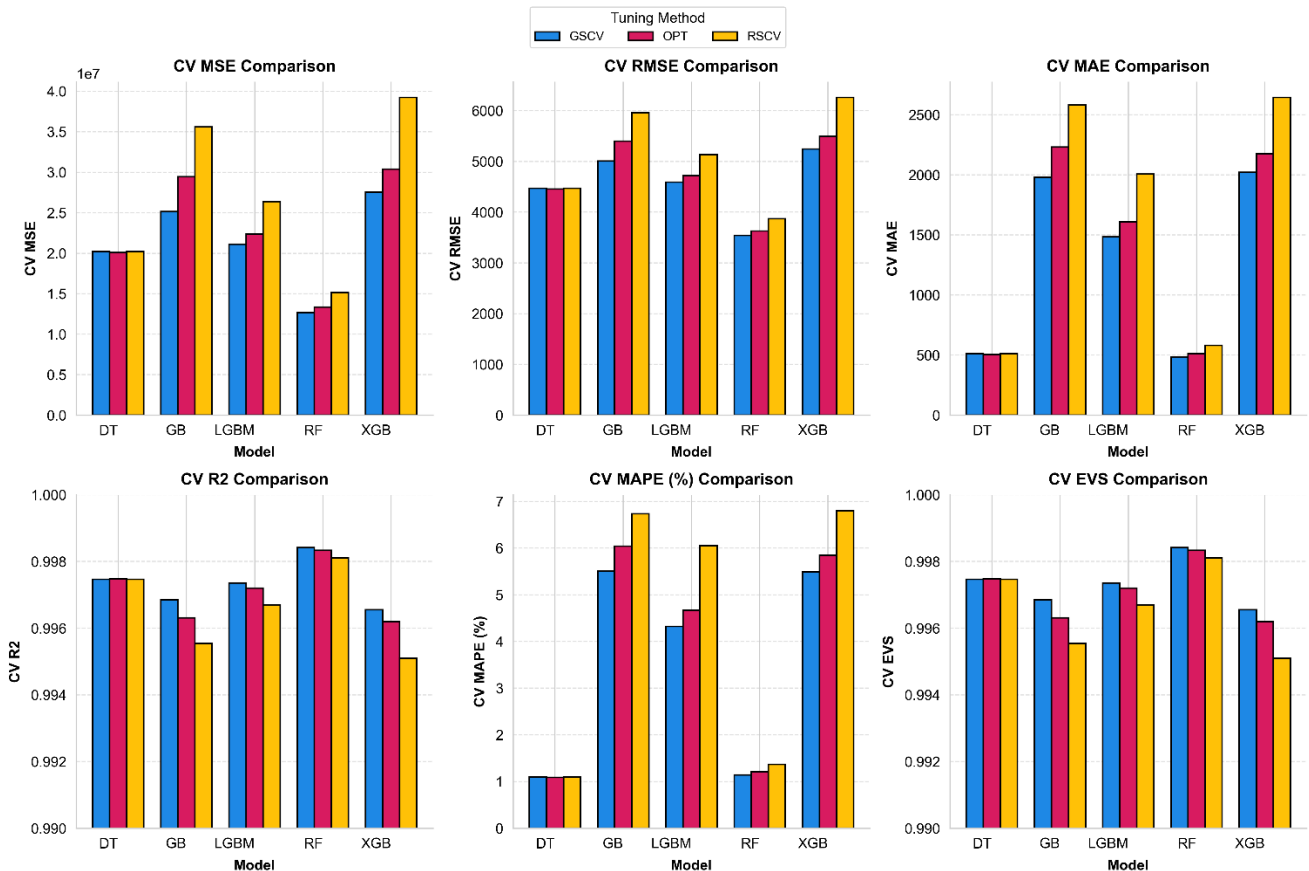


Figure 3. Performance comparison of hyperparameter-tuned regressors with CV results

Table 4. Performance comparison of models with CV results

Model	Tuning Method	CV MSE ($\times 10^7$)	CV RMSE	CV MAE	CV R ²	CV MAPE (%)	CV EVS
Decision Tree	GSCV	2.03	4476.60	511.86	0.997	1.10	0.997
	RSCV	2.03	4476.60	511.86	0.997	1.10	0.997
	OPT	2.06	4511.91	533.07	0.997	1.15	0.997
Random Forest	GSCV	1.27	3545.16	485.01	0.998	1.14	0.998
	RSCV	1.52	3874.45	579.83	0.998	1.37	0.998
	OPT	1.27	3551.13	484.89	0.998	1.14	0.998
Gradient Boosting	GSCV	2.51	5008.54	1977.93	0.997	5.50	0.997
	RSCV	3.56	5961.22	2579.51	0.996	6.73	0.996
	OPT	3.17	5611.47	2363.19	0.996	6.31	0.996
XGBoost	GSCV	2.75	5243.27	2020.61	0.997	5.50	0.997
	RSCV	3.92	6259.22	2642.84	0.995	6.80	0.995
	OPT	3.27	5694.66	2296.39	0.996	6.11	0.996
LightGBM	GSCV	2.11	4590.11	1482.74	0.997	4.32	0.997
	RSCV	2.64	5133.02	2005.39	0.997	6.05	0.997
	OPT	2.23	4720.88	1598.42	0.997	4.66	0.997

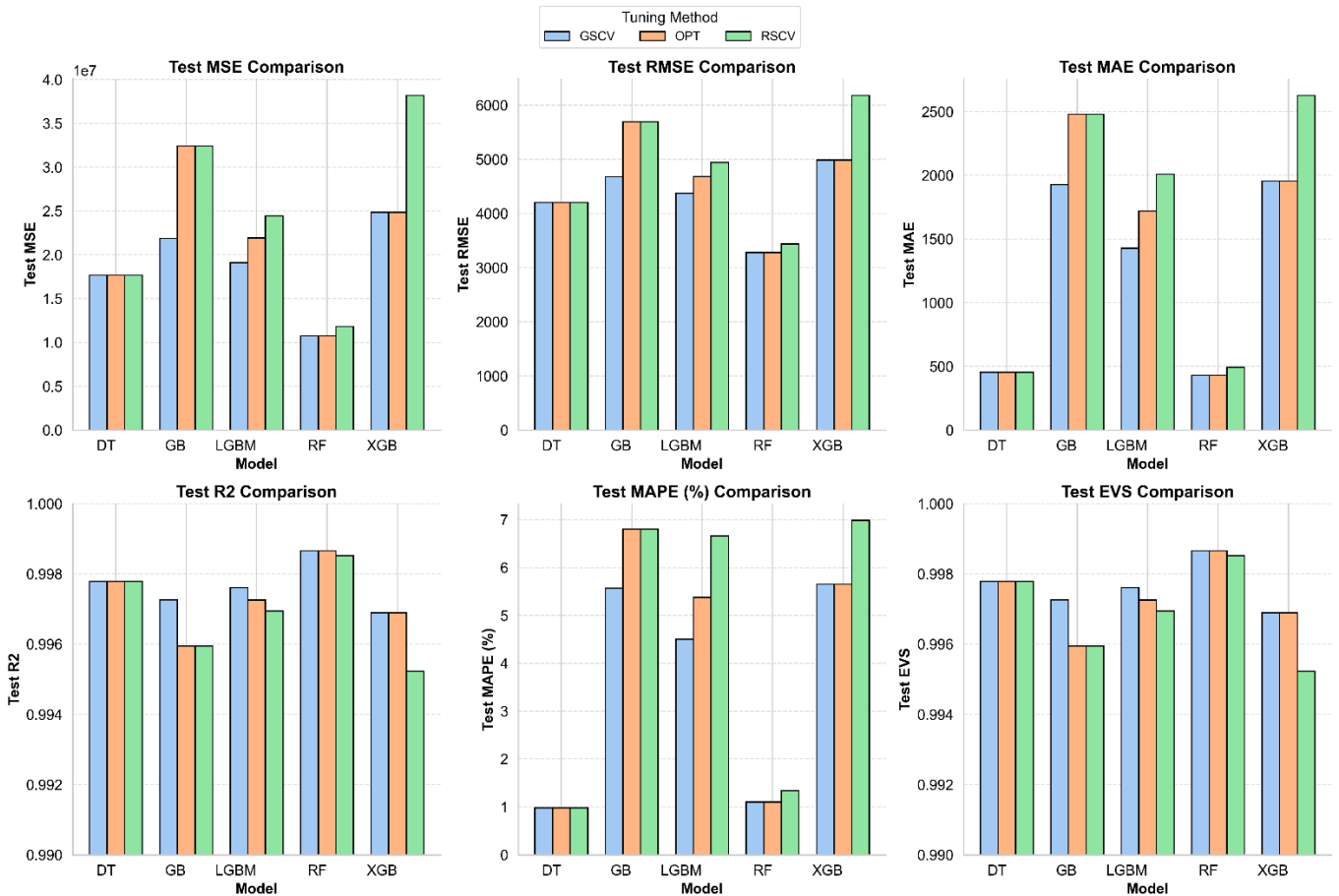


Figure 4. Performance comparison of hyperparameter-tuned regressors with test set results

Table 5. Performance comparison of models with test set results

Model	Tuning Method	Test MSE (×10 ⁷)	Test RMSE	Test MAE	Test R ²	Test MAPE (%)	Test EVS
Decision Tree	GSCV	1.77	4205.41	454.90	0.9978	0.9822	0.9978
	RSCV	1.77	4205.41	454.90	0.9978	0.9822	0.9978
	OPT	1.77	4205.41	454.90	0.9978	0.9822	0.9978
Random Forest	GSCV	1.08	3280.77	432.30	0.9987	1.1046	0.9987
	RSCV	1.18	3440.38	493.75	0.9985	1.3416	0.9985
	OPT	1.08	3280.77	432.30	0.9987	1.1046	0.9987
Gradient Boosting	GSCV	2.19	4678.22	1927.34	0.9973	5.5733	0.9973
	RSCV	3.24	5694.62	2481.35	0.9959	6.8060	0.9959
	OPT	3.24	5694.62	2481.35	0.9959	6.8060	0.9959
XGBoost	GSCV	2.49	4985.67	1955.69	0.9969	5.6556	0.9969
	RSCV	3.82	6178.80	2626.57	0.9952	6.9887	0.9952
	OPT	2.49	4985.67	1955.69	0.9969	5.6556	0.9969
LightGBM	GSCV	1.91	4373.84	1427.40	0.9976	4.5070	0.9976
	RSCV	2.45	4944.96	2009.55	0.9969	6.6615	0.9969
	OPT	2.19	4684.83	1720.47	0.9973	5.3834	0.9973

LGBM shows moderate and variable performance, with RMSE ranging from 4373.84 (GSCV) to 4944.96 (RSCV). DT yields consistent test outcomes across all tuning techniques (RMSE = 4205.41, R² = 0.9978), demonstrating minimal sensitivity to hyperparameter adjustments. Ensemble models benefit more from structured optimization, while simpler models show marginal gains.

5.3 Computational efficiency analysis

Computational cost is evaluated using tuning and training time (Table 3). GSCV incurs the highest cost due to exhaustive search. For instance, RF tuning time reaches 180.03 s under GSCV compared to 64.06 s under RSCV.

RSCV reduces tuning time by sampling a fixed number of configurations. There is a particularly noticeable difference in XGB, where tuning time reduces from 8.18 to 3.37 seconds (GSCV to RSCV) (58.8% reduction). OPT requires more time than RSCV but achieves accuracy comparable to GSCV. For example, RF tuning time increases to 228.16 s under OPT while maintaining identical accuracy to GSCV. Exhaustive search improves accuracy, whereas sampling-based methods improve efficiency.

5.4 Stability and robustness analysis

Stability of the models is assessed based on consistency across all folds and tuning methods. We found that the

Random Forest model is very consistent. Both GSCV and OPT resulted in a RMSE of 3280.77. This consistency indicates performance saturation and low sensitivity to hyperparameter changes. The RF model also showed identical performance across both methods. This clearly demonstrates a convergence to equivalent configurations, irrespective of the tuning approach employed. In contrast, boosting models show greater variability, particularly under RSCV.

5.5 Integrated trade-off analysis

The results show a trade-off among accuracy, computational cost, and stability. GSCV provides accurate and stable results at high computational cost. RSCV reduces tuning

time by up to 50-60% with a modest loss in accuracy. OPT achieves a balance by delivering accuracy comparable to GSCV with moderate computational cost.

RF tuned with GSCV or OPT achieves the best overall performance, while RSCV remains suitable for time-constrained settings. The choice of tuning strategy depends on the balance between accuracy, computational resources, and stability requirements.

Violin plots in Figure 5 further illustrates the distribution of cross-validation metrics across tuning methods which highlights differences in variability and stability. GSCV and OPT exhibit more consistent performance with narrower distributions, whereas RSCV shows greater variability, indicating sensitivity to random sampling.

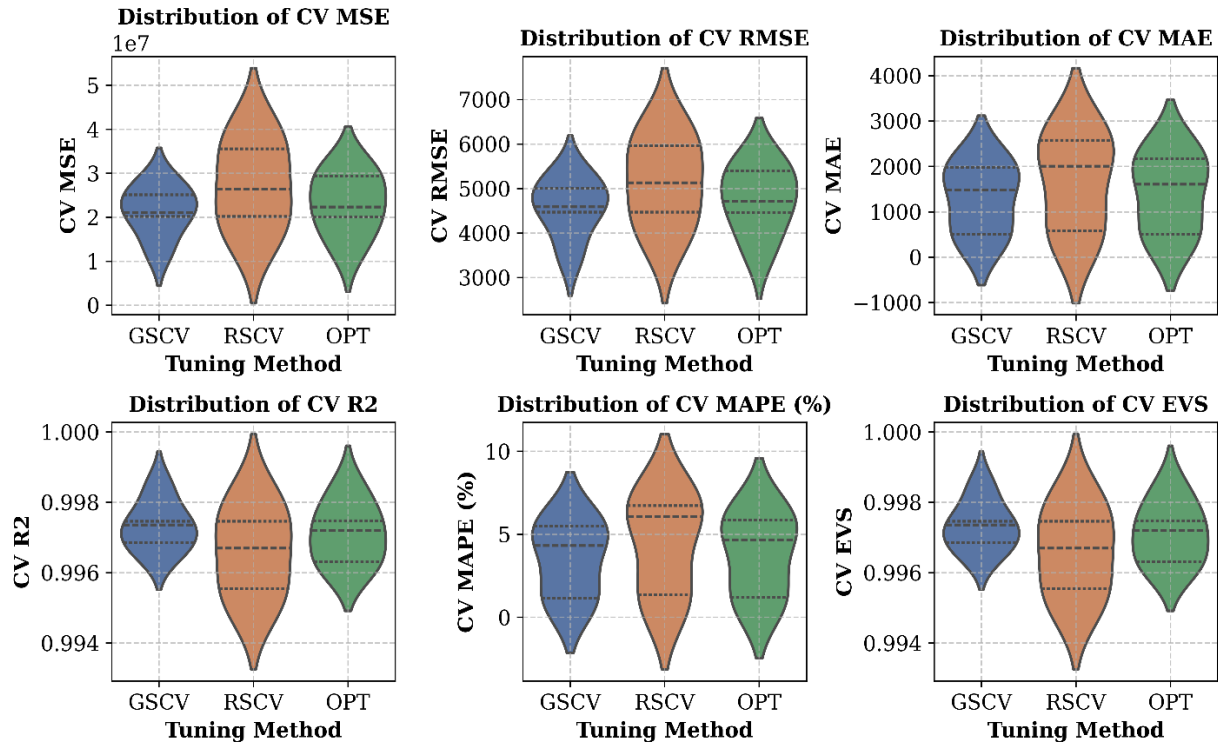


Figure 5. Distribution of cross-validation metrics across tuning methods

6. STATISTICAL ANALYSIS OF THE PERFORMANCE OF TUNING METHODS WITH CV

To assess statistical differences among hyperparameter tuning methods, the Shapiro-Wilk test was applied to cross-validation metrics (MSE, RMSE, MAE, R², MAPE, and EVS). All metrics exhibited non-normal distributions ($p < 0.001$), necessitating the use of the Friedman test. The Friedman test revealed a statistically significant difference among the tuning methods with a p-value of 0.0294. We also conducted pairwise comparisons using Wilcoxon signed-rank tests with Bonferroni correction, and the results are shown in Table 6.

Although the Friedman test indicates overall differences, the Wilcoxon tests do not reveal statistically significant pairwise differences after correction ($p > 0.05$). The smallest p-value is found between OPT and RSCV, which is 0.0625. However, this result is not statistically significant, as the adjusted p-value is 0.1875.

Table 6. Post-hoc Wilcoxon signed-rank test for cross-validation performance

Metric	Comparison	P-Value	Adjusted P-Value
MSE	GSCV vs OPT	0.12500	0.37500
	GSCV vs RSCV	0.12500	0.37500
	OPT vs RSCV	0.06250	0.18750
RMSE	GSCV vs OPT	0.12500	0.37500
	GSCV vs RSCV	0.12500	0.37500
	OPT vs RSCV	0.06250	0.18750
MAE	GSCV vs OPT	0.12500	0.37500
	GSCV vs RSCV	0.12500	0.37500
	OPT vs RSCV	0.06250	0.18750
R ²	GSCV vs OPT	0.12500	0.37500
	GSCV vs RSCV	0.12500	0.37500
	OPT vs RSCV	0.06250	0.18750
MAPE	GSCV vs OPT	0.12500	0.37500
	GSCV vs RSCV	0.12500	0.37500
	OPT vs RSCV	0.06250	0.18750
EVS	GSCV vs OPT	0.12500	0.37500
	GSCV vs RSCV	0.12500	0.37500
	OPT vs RSCV	0.06250	0.18750

These findings indicate that, while differences among tuning methods are observable, they are not statistically distinguishable on a pairwise basis given the limited sample size. The consistent patterns of p-values across different metrics indicate that the relative performance of the methods remains stable. The results indicate that no single tuning method is the best choice for every situation. Method selection should be driven by trade-offs between computational cost and stability. To further interpret the predictive behavior of the models and understand the contribution of input features, a SHAP-based interpretability analysis is performed.

7. SHAP-BASED FEATURE IMPORTANCE ANALYSIS

A SHAP analysis was conducted to enhance model interpretability and understand the impact of individual features on CYP. Although RF demonstrated the best

predictive performance, SHAP analysis was conducted using the XGB model, particularly, due to its efficient integration with TreeSHAP, which enables fast and consistent computation of feature contributions. Furthermore, both RF and XGB are tree-based ensemble models that capture similar feature interactions. Therefore, we performed the interpretability analysis with XGB to understand the overall modeling behavior. The effect of various features on crop yield is illustrated with the SHAP summary plot (Figure 6). Among environmental factors, pesticides exhibited the most decisive influence, indicating their substantial impact on yield prediction. To assess the influence of features on CYP, SHAP analysis of the XGB model unveiled Crop type (item) as the most influential feature. It suggests that crop yield varies from one crop type to another. Then, among the other factors, pesticide usage followed by temperature were in an important sequence. Rainfall, Area, and Year also slightly contributed to CYP.

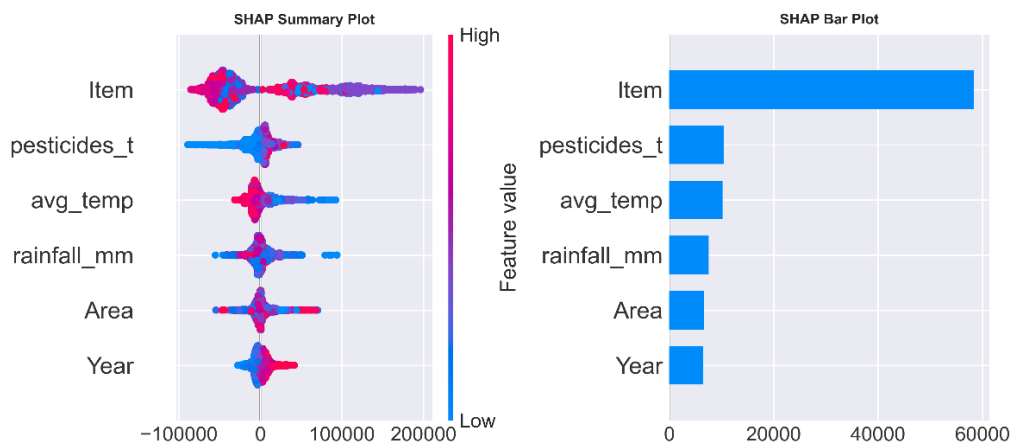


Figure 6. SHAP analysis showing (a) summary plot of individual prediction impacts for each feature and (b) bar plot of overall mean absolute SHAP values indicating feature importance

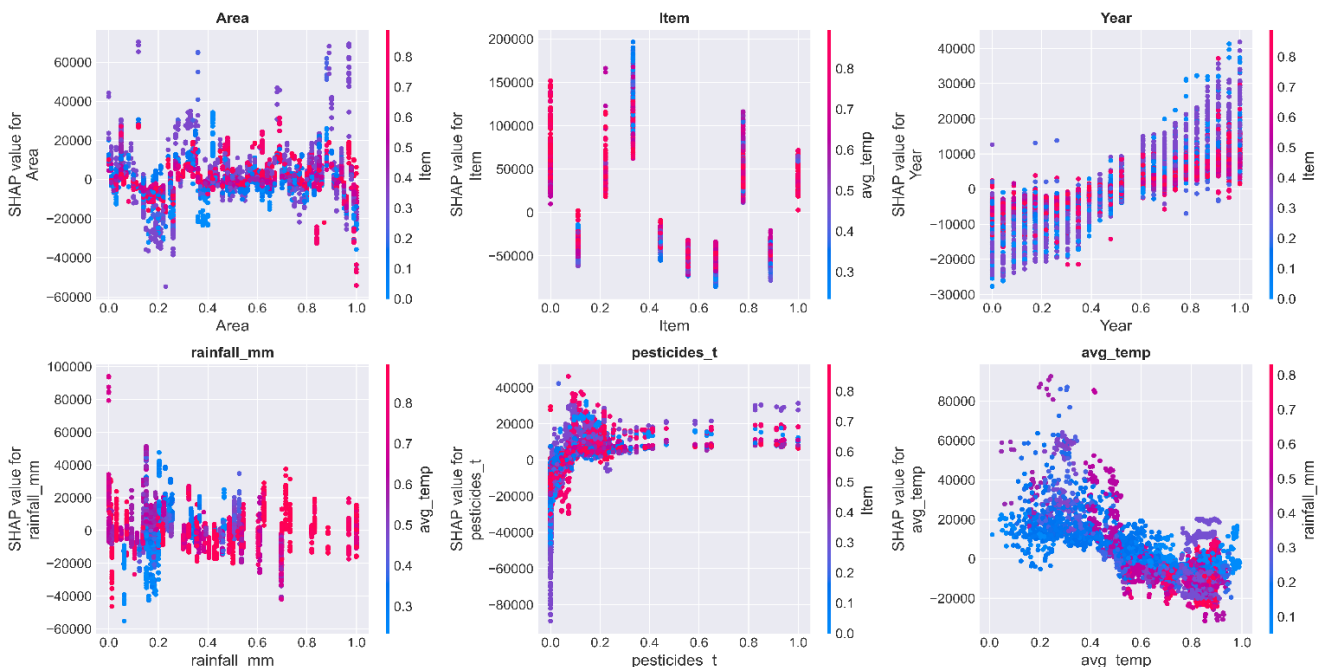


Figure 7. SHAP dependence plots illustrating the interaction between key input features and their corresponding SHAP values for crop yield prediction (CYP)

Note: Each subplot shows the relationship between the normalized values of features (x-axis) and its SHAP value (y-axis). Color gradients specify the value of an interacting feature, capturing possible interaction and non-linear effects

SHAP dependency plots shown in Figure 7 give more insight into the interactions between features. The figure shows that Temperature has a non-linear inverse correlation with SHAP value. So, it is evident that crop yield is sensitive to crop Temperature variations. Pesticides showed a saturation effect in their interactions. This implies the prevalence of underlying temporal trends in agricultural practices or climate patterns. These findings are consistent with the predictive performance results, where ensemble models demonstrated strong sensitivity to agronomic and environmental variables.

8. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this study, we investigated how GSCV, RSCV, and OPT hyperparameter tuning methods affect regression models in CYP. We observed that the performance of RF was the most consistent across cross-validation and test evaluations. When tuned with GSCV and OPT, it had lower error values and higher goodness-of-fit than the other models.

A different trend was observed for boosting-based models. In particular, XGB and GB showed noticeable variation in performance depending on the tuning approach. Under RSCV, both models demonstrated increased prediction error. This suggests a higher sensitivity to suboptimal parameter configurations. DT performance did not change much with different tuning methods, indicating low reliance on hyperparameter selection.

From a computational perspective, a clear trade-off between predictive accuracy and efficiency was observed. For instance, in RF tuning of GSCV (180 seconds) and OPT (228 seconds) took substantially longer than RSCV (64 seconds). The absence of statistically significant pairwise differences further supports that performance variations are not universally decisive. Based on these observations, tuning strategies must trade off computational efficiency and predictive stability.

Future research directions include:

- Developing scalable search techniques, highlighting the challenge of increasing data size.
- Expounding Cost-sensitive tuning techniques that dynamically adjust search space sizes according to available resources and model complexity.
- Performing Generalization by examining how different hyperparameter tuning approaches cooperate with the model's capacity to transfer to diverse data distributions.
- Studying AutoML frameworks for efficient and reproducible tuning with reduced human effort.
- Extend the proposed framework in real-time processing and thereby explore solutions for delays and changing environmental conditions.

The findings of this research will help developers in selecting suitable hyperparameter optimization methods more efficiently and effectively. Thereby, this will be useful in developing efficient agricultural decision support systems for agrarian practitioners, which may lead to optimized crop productivity through more reliable yield predictions.

REFERENCES

[1] Suprajitno, H. (2023). Long-term forecasting of crop water requirement with BP-RVM algorithm for food

security and harvest risk reduction. *International Journal of Safety & Security Engineering*, 13(3): 565-575. <https://doi.org/10.18280/ijssse.130319>

[2] Halder, M., Datta, A., Siam, M.K.H., Mahmud, S., Sarkar, M.S., Rana, M.M. (2023). A systematic review on crop yield prediction using machine learning. In *Intelligent Systems and Networks*, Springer, Singapore, pp. 658-667. https://doi.org/10.1007/978-981-99-4725-6_77

[3] Nguyen, H., Randall, M., Lewis, A. (2024). Factors affecting crop prices in the context of climate change - A review. *Agriculture*, 14(1): 135. <https://doi.org/10.3390/agriculture14010135>

[4] Lobell, D.B., Burke, M.B. (2010). On the use of statistical models to predict crop yield responses to climate change. *Agricultural and Forest Meteorology*, 150(11): 1443-1452. <https://doi.org/10.1016/j.agrformet.2010.07.008>

[5] Tamasiga, P., Onyeaka, H., Bakwena, M., Happonen, A., Molala, M. (2023). Forecasting disruptions in global food value chains to tackle food insecurity: The role of AI and big data analytics - A bibliometric and scientometric analysis. *Journal of Agriculture and Food Research*, 14: 100819. <https://doi.org/10.1016/j.jafr.2023.100819>

[6] Pandey, D.K., Mishra, R. (2024). Towards sustainable agriculture: Harnessing AI for global food security. *Artificial Intelligence in Agriculture*, 12: 72-84. <https://doi.org/10.1016/j.aiaa.2024.04.003>

[7] Abbas, F., Afzaal, H., Farooque, A.A., Tang, S. (2020). Crop yield prediction through proximal sensing and machine learning algorithms. *Agronomy*, 10(7): 1046. <https://doi.org/10.3390/agronomy10071046>

[8] Dhal, S.B., Kar, D. (2024). Transforming agricultural productivity with AI-driven forecasting: Innovations in food security and supply chain optimization. *Forecasting*, 6(4): 925-951. <https://doi.org/10.3390/forecast6040046>

[9] Bischl, B., Binder, M., Lang, M., Pielok, T., et al. (2021). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2): e1484. <https://arxiv.org/abs/2107.05847>.

[10] Jones, E.J., Bishop, T.F., Malone, B.P., Hulme, P.J., Whelan, B.M., Filippi, P. (2022). Identifying causes of crop yield variability with interpretive machine learning. *Computers and Electronics in Agriculture*, 192: 106632. <https://doi.org/10.1016/j.compag.2021.106632>

[11] Cedric, L.S., Adoni, W.Y.H., Aworka, R., Zoueu, J.T., Mutombo, F.K., Krichen, M., Kimpolo, C.L.M. (2022). Crops yield prediction based on machine learning models: Case of west African countries. *Smart Agricultural Technology*, 2: 100049. <https://doi.org/10.1016/j.atech.2022.100049>

[12] Naik, N.K., Sethy, P.K., Panigrahi, M., Behera, S.K. (2023). Support vector machine classifier for wheat grain identification based on grid search optimization technique. In *International Conference on ICT for Sustainable Development*, Springer, Singapore, pp. 237-245. https://doi.org/10.1007/978-981-99-5652-4_22

[13] Abdel-Salam, M., Kumar, N., Mahajan, S. (2024). A proposed framework for crop yield prediction using hybrid feature selection approach and optimized machine learning. *Neural Computing and Applications*, 36(33):

- 20723-20750. <https://doi.org/10.1007/s00521-024-10226-x>
- [14] Pedregosa, F., Varoquaux, G., Gramfort, A., Michelet, V., et al. (2011). Scikit-learn: Machine learning in Python. arXiv Preprint arXiv:1201.0490. <https://doi.org/10.48550/arXiv.1201.0490>
- [15] Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13: 281-305.
- [16] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, Tokyo, Japan, pp. 2623-2631. <https://doi.org/10.1145/3292500.3330701>
- [17] Jayanthi, S., Sathya, M.A.J., Nathan, B., Karmokanda, K. (2025). Ensemble learning framework for crop yield prediction with optuna hyperparameter tuning. *Journal of Information Systems Engineering and Management*, 10: 783-795. <https://doi.org/10.52783/jisem.v10i27s.4553>
- [18] Song, Y., Zhan, D., He, Z., Li, W., Duan, W., Yang, Z., Lu, M. (2023). HPO-empowered machine learning with multiple environment variables enables spatial prediction of soil heavy metals in coastal delta farmland of China. *Computers and Electronics in Agriculture*, 213: 108254. <https://doi.org/10.1016/j.compag.2023.108254>
- [19] Upreti, K., Lingareddy, N., Deepika, S., Kumar, N., Parashar, J., Divakaran, P. (2024). Optimization ensemble learning techniques for reliable using ML. In *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)*, Bali, Indonesia, pp. 431-436. <https://doi.org/10.1109/TIACOMP64125.2024.00078>
- [20] Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J. (2017). *Classification and Regression Trees*. Chapman and Hall/CRC. <https://doi.org/10.1201/9781315139470>
- [21] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1): 5-32. <https://doi.org/10.1023/A:1010933404324>
- [22] Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189-1232. <https://doi.org/10.1214/aos/1013203451>
- [23] Chen, T., Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, NY, USA, pp. 785-794. <https://doi.org/10.1145/2939672.2939785>
- [24] Ke, G., Meng, Q., Finley, T., Wang, T., et al. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30: 3146-3154.
- [25] Aderele, M.O., Srivastava, A.K., Butterbach-Bahl, K., Rahimi, J. (2025). Integrating machine learning with agroecosystem modelling: Current state and future challenges. *European Journal of Agronomy*, 168: 127610. <https://doi.org/10.1016/j.eja.2025.127610>
- [26] Dastres, E., Edalat, M. (2025). Deep ensemble learning for weed risk mapping: Hybrid RF-CatBoost and CNN-XGBoost algorithms for predicting *Chenopodium album* distribution in rapeseed fields. *Smart Agricultural Technology*, 101731. <https://doi.org/10.1016/j.atech.2025.101731>
- [27] Huber, F., Yushchenko, A., Stratmann, B., Steinhage, V. (2022). Extreme gradient boosting for yield estimation compared with deep learning approaches. *Computers and Electronics in Agriculture*, 202: 107346. <https://doi.org/10.1016/j.compag.2022.107346>
- [28] Ansarifar, J., Wang, L., Archontoulis, S.V. (2021). An interaction regression model for crop yield prediction. *Scientific Reports*, 11(1): 17754. <https://doi.org/10.1038/s41598-021-97221-7>
- [29] Varma, S., Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1): 91. <https://doi.org/10.1186/1471-2105-7-91>
- [30] Chlingaryan, A., Sukkarieh, S., Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture*, 151: 61-69. <https://doi.org/10.1016/j.compag.2018.05.012>
- [31] Lundberg, S.M., Lee, S.I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
- [32] Ribeiro, M.T., Singh, S., Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, NY, USA, pp. 1135-1144. <https://doi.org/10.1145/2939672.2939778>